

NPI Retrieve

Guide til Anvendere

Indhold

1	Introduktion	3
1.1	Formål.....	3
1.2	Læsevejledning.....	3
1.3	Dokumenthistorik	3
1.4	Definitioner og referencer.....	3
2	Anvendelse af NPI Udtræksservice	4
2.1	Forberedelse til kald af NPI Udtræksservice.....	4
2.1.1	Indhentning af Dokument-metadatainformation til udtræk	5
2.2	Kald af NPI Udtræksservice	5
3	Eksempler på kald af NPI Udtræksservice	5
3.1	Kildesystem indhenter DGWS ID-kort fra STS	5
3.2	Anvendersystem skaber HSUID-header.....	6
3.3	Anvendersystem skaber RetrieveDocumentSetRequestType	7
3.4	Anvendersystem kalder RetrieveDocumentSet	8
3.5	Klientgenerering ud fra WSDL.....	8

1 Introduktion

1.1 Formål

Dette dokument beskriver anvendelsen af den webservice, der stilles til rådighed af NPI Udtræksservice.

Dokumentet giver en introduktion til det generelle webserviceinterface samt et eksempel på hvorledes den kan anvendes i et specifikt anvendelsesystem.

1.2 Læsevejledning

Dette dokument henvender sig til udviklere og arkitekter der skal anvende NPI Udtræksservice på NSP. Det antages at læseren er bekendt med webservices samt brug af den gode web service (DGWS) og Security Token Service (STS).

1.3 Dokumenthistorik

Version	Dato	Ansvarlig	Beskrivelse
0.1	14.01.2013	Systematic	Initiel udgave
0.9	27.2.2013	Systematic	Klargjort til afprøvning i demonstrationsprojektet.
0.9a	18.4.2013	Systematic	Afsnit 2.1.1: homeCommunityId læses fra home-attribute.

1.4 Definitioner og referencer

Definition	Beskrivelse
NPI	Nationalt Patient Indeks
NSI	National Sundheds-IT
NSP	Den nationale service platform (inden for sundheds-IT)
SHAK	Sygehusafdelingsklassifikation
SOR	Sundhedsvæsenets organisationsregister
STS	Security Token Service

Alias	Beskrivelse
DGWS 1.0	Den Gode Webservice 1.0
DGWS 1.0.1	Den Gode Webservice 1.0.1
HSUID-header	Healthcare Service User Identification Header (SSE/11734/IFS/0006)
IHE2	IHE Current Technical Framework – Revision 8.0, Vol. 2a: Transactions (ITI1-ITI28) http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_Rev8-0_Vol2a_FT_2011-08-19.pdf
IHE3	IHE Current Technical Framework – Revision 8.0, Vol. 2b: Transactions (ITI29-ITI64)

Alias	Beskrivelse
	http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_Rev8-0_Vol2b_FT_2011-08-19.pdf
Metadata	NPI Metadata Datamodel (SSE/11734/SDD/0001)
NPI opslag anvenderguide	NPI Service, Guide til anvendere (SSE/11734/PHB/0009)
NPI opslag ws	NPI Service snitfladebeskrivelse (SSE/11734/IFS/0001)
NPI Retrieve ws	NPI Udtræk snitfladebeskrivelse (SSE/11734/IFS/0009)

2 Anvendelse af NPI Udtræksservice

NPI Udtræksservice udstiller en operation til udtræk af dokument på baggrund af metadata fremfundet i Det Nationale Patient Indeks via en webservice-snitflade beskrevet i [NPI Retrieve ws].

Udtræk foretages under anvendelse af standarden IHE Cross-Enterprise Document Sharing (XDS), der er baseret på ebXML Registry Services (RS) og ebXML Registry Information Model (RIM).

Udtræksoperationen er i IHE XDS-termer RetrieveDocumentSet-forespørgslen i IHE Technical Framework's transaktion 43, Retrieve Document Set.

Med henblik på at være baseret på Retrieve Document Set i videst muligt omfang, er ændringer til IHE XDS-standardens WSDL minimeret. Derfor er payload til operationen uændret.

Som beskrevet i [NPI Retrieve ws] forventer NPI Udtræksservice, at Retrieve Document Set-kaldet indeholder ekstra headere i form af:

- En Security-header
- En Medcom-header
- En Healthcare Service User Identification (HSUID) header.

De første to kommer af Den Gode Webservice beskrevet i [DGWS 1.0] og [DGWS 1.0.1], mens den sidste er beskrevet i [HSUID-header].

Headerne er tilføjet som implicitte headere, dvs. ved at være beskrevet i WSDL'ens bindings uden at være tilføjet til Retrieve Document Set SOAP body-strukturen.

2.1 Forberedelse til kald af NPI Udtræksservice

Inden kald af NPI Service er det anvendersystemets ansvar:

- at sikre, at forudsætninger for anvendelse som beskrevet i [NPI Retrieve ws] er opfyldt, fx at værdispring kun gennemføres af sundhedspersoner, der er autoriseret til det.
- at indhente et ID-kort på krævet sikkerhedsniveau udstedt af STS

Bemærk, at anvendersystemets CVR-nummer, som indlejres i ID-kortet, skal være at finde på whitelisen for NPI Udtræksservice. Tilføjelse på denne whitelist skal aftales med systemejere for NPI Udtræksservice.

2.1.1 Indhentning af Dokument-metadatainformation til udtræk

Inden der udtrækkes dokumenter skal der indhentes metadata om de dokumenter, der skal udtrækkes. Til dette formål anvendes NPI opslag beskrevet i [NPI opslag ws] og [NPI opslag anvenderguide].

Ved opslag specificeres søgeparametre, herunder id for den patient, der ønskes dokument-metadata om.

Svaret fra NPI, `AdhocQueryResponse`, kan indeholde flere dokument-metadataobjekter i `/AdhocQueryResponse/RegistryObjectList`. For hvert dokument-metadataobjekt, hvor tilhørende dokument ønskes udtrukket, fremfindes det i `/AdhocQueryResponse/RegistryObjectList`. De værdier, der skal anvendes i udtræk, fremfindes under dokument-metadataobjektet `ExtrinsicObject` som følger:

- `DocumentRequest.documentUniqueId`: værdien af `ExtrinsicObject/ExternalIdentifier`, hvor attributten `identificationScheme` er sat til `"urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"`.
- `DocumentRequest.repositoryUniqueId`: værdien af `ExtrinsicObject/Slot`, hvor attributten `name` er sat til `"repositoryUniqueId"`.
- `DocumentRequest.homeCommunityId`: værdien af attributten `home` på `ExtrinsicObject`'et. Værdien sættes kun, når attributten er defineret på `ExtrinsicObject`'et.

For hvert dokumenter, der ønskes udtrukket, skabes en `DocumentRequest`-struktur, som tilføjes til request'en. Se yderligere detaljer i afsnit 3.3.

2.2 Kald af NPI Udtræksservice

Ved kald af operationen `DocumentRepository_RetrieveDocumentSet` på NPI Udtræksservice, skal følgende gennemføres:

1. Der skabes en `RetrieveDocumentSetRequestType` (dvs. payload af `DocumentRepository_RetrieveDocumentSet`) skal populeres
2. Security-header populeres med ID-kortet fra STS
3. Medcom-header skabes, herunder:
 - a. Skal der tildeles unikt besked-id
 - b. Skal der tildeles unikt sessionsid til Medcom-headerens `flowID`-element
4. HSUID-header skabes, og tildeles værdier til beskrivelse af den person, der foretager kaldet, henholdsvis beskrivelse af patienten

3 Eksempler på kald af NPI Udtræksservice

Dette afsnit giver et eksempel på, hvordan denne service kan kaldes.

Der er pt. ikke skrevet .NET-eksempelkode på at kalde NPI Udtræksservice, men der er WSDL'er til rådighed, som man, vha. Visual Studio's webservice wizard kombineret med Seal.NET (se herunder), kan genere en web service consumer ud fra.

3.1 Kildesystem indhenter DGWS ID-kort fra STS

I det følgende skitseres, hvordan et kildesystem udviklet i Java kan benytte Seal.Java til at indhente et STS-signeret ID-kort.

Seal er et open source framework til understøttelse af "Den Gode Webservice", herunder integration til en central identitetsservice (IdP/STS), håndtering af føderationscertifikater mv.

Det findes i udgaver til Java og .Net.

Komponent	Reference
Seal.Java	http://digitaliser.dk/group/374971
Seal.NET	http://digitaliser.dk/group/375117

Indhentning af et STS-signeret ID-kort sker under anvendelse af et virksomhedscertifikat som skitseret ved følgende:

```
Properties properties = SignatureUtil.setupCryptoProviderForJVM();

// Her kan anvendes new SOSITestFederation(properties), hvis test-miljø anvendes
Federation federation = new SOSIFederation(properties);

File keystoreFile = ... // read in the keystore file
String keystorePassword = ... // obtain the password to the keystore file
CredentialVault vault = new FileBasedCredentialVault(
    properties,
    keystoreFile,
    keystorePassword);

SOSIFactory factory = new SOSIFactory(federation, vault, properties);

String flowId = ... // Et unikt flow-id, der sættes i ID-kortet

SystemIDCard systemIDCard = factory.createNewSystemIDCard(
    ...); // Her angives forskellige parametre til indhold i ID-kortet

SecurityTokenRequest securityTokenRequest = factory.createNewSecurityTokenRequest();
securityTokenRequest.setIDCard(systemIDCard);
SecurityTokenResponse response = ... // Send securityTokenRequest til STS'en og deserialiser svaret
// med factory.deserializeSecurityTokenResponse
...
IDCard signedIDCard = response.getIDCard();
```

Hermed er indhentet et ID-kort signeret af STS'en. I nedenstående eksempler er det antaget, at friske requests til kald af de respektive webservice-operationer er skabt; at det signerede ID-kort er kopieret over i den friske request; at øvrige DGWS-parametre sættes. Dette er i nedenstående beskrevet som en hjælpemetode `getValidDGWSHeaders`, som anvender antages at lave.

```
DGWSHeaderWrapper signedHeaders = getValidDGWSHeaders(flowId);
```

3.2 Anvendersystem skaber HSUID-header

```
// Create HSUID header
dk.nsi.hsuid._2013._01.hsuid_1_1.ObjectFactory objectFactory = new
    dk.nsi.hsuid._2013._01.hsuid_1_1.ObjectFactory();
HsuidHeader hsuidHeader = objectFactory.createHeader();
```

```

// Create assertion
AssertionType assertion = objectFactory.createAssertionType();
Date now = new Date();
Calendar calendar = Calendar.getInstance(TimeZone.getTimeZone("UTC"));
calendar.setTimeInMillis(now.getTime());
assertion.setIssueInstant(calendar);
assertion.setIssuer(" ... "); // Replace with name of my organisation
assertion.setVersion(BigDecimal.valueOf(2.0));
assertion.setId(AssertionTypeId.HSUID);
hsuidHeader.setAssertion(assertion);

// Create attribute statement
AttributeStatement attributeStatement = objectFactory.createAttributeStatement();
attributeStatement.setId(AttributeStatementId.HSUIDdata);

// Create attribute for acting user (Danish: bruger)
dk.nsi.hsuid._2013._01.hsuid_1_1.Attribute actingUserAttribute = objectFactory.createAttribute();
actingUserAttribute.setName("nsi:ActingUserCivilRegistrationNumber");
actingUserAttribute.setAttributeValue(" ... "); // Indsæt cpr-nummer uden bindestreg for bruger
attributeStatement.getAttribute().add(actingUserAttribute);

// Create attribute for acting user's organization
dk.nsi.hsuid._2013._01.hsuid_1_1.Attribute orgUsingAttribute = objectFactory.createAttribute();
orgUsingAttribute.setName("nsi:OrgUsingID");
// Here, the acting user's organization is set to 12345678 in the SOR classification.
// Use name format "nsi:ynumber" and "nsi:skskode", respectively, for organization stated
// in classifications for general practioner and hospital departments (respectively)
orgUsingAttribute.setNameFormat(SubjectIdentifierType.valueOf("nsi:sor"));
orgUsingAttribute.setAttributeValue("12345678");
attributeStatement.getAttribute().add(orgUsingAttribute);

... // Med tilsvarende kode sættes øvrige attributter i HSUID-header

assertion.setAttributeStatement(attributeStatement);

```

Den skabte hsuidHeader er nu klar til anvendelse.

3.3 Anvendersystem skaber RetrieveDocumentSetRequestType

Til request'en skabes en RetrieveDocumentSetRequestType-struktur indeholdende identifikation af et til flere dokumenter.

```

// For hvert ønsket dokument fremfindes værdier som beskrevet ovenfor
String homeCommunityId = " ... ";
String repositoryUniqueId = " ... ";
String documentUniqueId = " ... ";

ihe.iti.xds_b._2007.ObjectFactory objFactory = new ihe.iti.xds_b._2007.ObjectFactory();
RetrieveDocumentSetRequestType request = objFactory.createRetrieveDocumentSetRequestType();
DocumentRequest documentRequest =
objFactory.createRetrieveDocumentSetRequestTypeDocumentRequest();
documentRequest.setHomeCommunityId(homeCommunityId);
documentRequest.setRepositoryUniqueId(repositoryUniqueId);
documentRequest.setDocumentUniqueId(documentUniqueId);
request.getDocumentRequest().add(documentRequest);

```

3.4 Anvendersystem kalder RetrieveDocumentSet

Oprettelse af service og port:

```
URL wsdl = new URL(" ... "); // Point to WSDL file
DocumentRepositoryService service =
    new DocumentRepositoryService(wsdl,
        new QName("urn:ihe:iti:xds-b:2007",
            "DocumentRepository_Service"));
DocumentRepositoryPortType port =
    service.getDocumentRepositoryPortSoap12(new MTOMFeature()); // Enable MTOM
```

Herefter kan metoden på porten kaldes:

```
HsuidHeader hsuidHeader = ... // Skabes som beskrevet ovenfor
RetrieveDocumentSetRequestType request = ... // Skabes som beskrevet ovenfor

String flowId = XMLUtil.createNonce(); // Et unikt flow-id, der sættes i ID-kortet, skabes vha. SEAL
DGWSHeaderWrapper signedHeaders = getValidDGWSHeaders(flowId); // se ovenfor

RetrieveDocumentSetResponseType response =
    port.documentRepositoryRetrieveDocumentSet(
        request,
        signedHeaders.getSecurityHeader(),
        new Holder<Header>(signedHeaders.getMedcomHeader()),
        hsuidHeader);
```

Dokumentet/dokumenterne kan nu udlæses som bytes i response, forudsat kaldet var succesfuldt:

```
if ("urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success".equals(
    response.getRegistryResponse().getStatus()) {
    for (RetrieveDocumentSetResponseType.DocumentResponse documentEntry:
        response.getDocumentResponse()) {
        if ("text/xml".equals(documentEntry.getMimeType()) { // handle text/xml
            byte[] documentBytes = documentEntry.getDocument();
            ... // handle the bytes
        }
        else if (...) { ... } // handle other MIME types
    }
}
```

3.5 Klientgenerering ud fra WSDL

En webservice-klient kan genereres med udgangspunkt i NPI Udtræksservice WSDL-filen, se [NPI Retrieve ws]. Det er ikke givet, at webservice-klientens proxy-klasser kan generes på baggrund af en deployed service (via ?wsdl-tilføjelse på servicen), da der ikke nødvendigvis er åbnet for sådanne kald.